REMARKS

In response to the Final Office Action mailed September 21, 2004, Applicants respectfully request reconsideration. Claims 1-15 were previously pending in this application. Applicants have amended claims 1 and 10 and submit new claims 16-18.

The Office Action rejected claims 1-5 and 8-10 under 35 U.S.C. §103(a) as purportedly being obvious over the combination of Song (5,546,599), Grochowski (6,353,883) and Hennessy (Hennessy and Patterson, Computer Architecture – A Quantitative Approach, 2nd Edition, 1996). The Office Action further rejected claims 6, 7, and 11-15 under 35 U.S.C. §103(a) as purportedly obvious over the combination of Song, Grochowski, and Hennessy, further in view of various other references. For the reasons discussed below, each of these rejections is respectfully traversed and the application as presented is believed to be in condition for allowance.

In Applicants' response mailed June 28, 2004, Applicants pointed out that Song teaches away from the idea of not decoding an instruction if an exception occurs in the fetch stage, as Song discloses that for instruction fetch and decode related exceptions, the processor detects such exceptions at the fetch or decode stage and, in response, dispatches the instruction that causes such an exception to a reservation stage of the execution unit. In view of this, Applicants pointed out that it would not have been obvious to modify Song to prevent an instruction from being supplied to the execution unit if an exception occurs. *See* Applicants' June 28, 2004 response, page 16.

The Final Office Action responds to this argument at page 16, paragraph 26, asserting that Hennessy teaches, at Figure 3.41 on page 184, that exceptions may occur when an instruction is to be fetched. The Office Action further asserts that, "[a] person would have realized that if a page fault exception occurs during the fetch, then fetch has failed and the instruction that is supposed to be fetched will not be decoded **until after the exception is fixed** (emphasis in original)." While Applicants do not deny that in the case of a page fault exception, the instruction that is to be fetched will not be decoded (nor will it be fetched), the breakpoint recited in claim 1 is different from a page fault. To clarify this point, Applicants have amended claim 1 to indicate that the breakpoint is caused by an instruction having a specified program count or an instruction having a specified opcode. This is different from a page fault, as a page

fault is caused by requested data not being stored in main memory, requiring that the data be fetched
from secondary memory (i.e., disk). A page fault is not caused by detection of an instruction with a certain opcode or detection of an instruction at a certain point in the program count.

Applicants have also amended claim 10 to clarify that the detecting of instructions which have a debug effect is based on a program count or an opcode of the instructions. As should be clear from the discussion above, this is very different from a page fault, as a page fault is caused by requested data not being stored in main memory and not by detection of an instruction with a certain opcode or detection of an instruction at a certain point in the program count.

At paragraph 27 on page 18, the Office Action argues that floating point exceptions, as disclosed by Song, and a breakpoint are the same, asserting that, "the exception is a break in normal program execution in order to fix an error or malfunction. In this broad sense, an exception is a break point, i.e., a point of breaking of normal execution for error correction. In addition, Hennessy, on page 180, has shown that a breakpoint is an exception." As stated above, Applicants have amended claims 1 and 10 to clearly distinguish over a floating point exception. A floating point exception is an exception that occurs when, for example, an operation attempts to divide a floating point number by zero or perform a calculation that exceeds the number of bits allocated to store the result. This is different from a breakpoint caused by an instruction having a specified program count or an instruction having a specified opcode, as recited in claim 1, and from detecting instructions which have a debug effect, based on a program count or an opcode of the instructions, as recited in claim 10.

In view of the foregoing, claims 1 and 10 patentably distinguish over the combination of Song, Grochowski, and Hennessy. Accordingly, it is respectfully requested that the rejections of claims 1 and 10 under 35 U.S.C. §103(a) be withdrawn.

Claims 2-9 depend from claim 1 and are patentable for at least the same reasons as claim 1. Claims 11-15 depend from claim 10 and are patentable for at least the same reasons as claim 10. Accordingly, it is respectfully requested that the rejection of these claims be withdrawn.

Applicants have added new claim 16, which is claim 5 as pending prior to this amendment, rewritten in independent form. Claim 16 is directed to a computer system for

executing predicated instructions wherein each instruction includes a guard, the value of which determines whether or not that instruction is executed. The computer system comprises: a fetch unit for fetching instructions to be executed; a decode unit for decoding said instructions; at least one pipelined execution unit for executing decoded instructions and being associated with a guard register file holding values of the guards to allow resolution of the guards to be made; and an emulation unit including control circuitry which cooperates with the decode unit to selectively control the decode unit to implement a precise watch or a non-precise watch on detection of a breakpoint wherein according to a precise watch, the instruction causing the breakpoint is not decoded by the decode unit and, according to a non-precise watch, the instruction causing the breakpoint and subsequent instructions are permitted to be supplied from the decode unit to the at least one execution unit while guard resolution in said at least one execution pipeline is awaited, wherein, when the emulation unit is in the precise watch mode, it is operable to issue a request to the execution pipeline for guard resolution, the guard resolution being transmitted to the control circuitry of the emulation unit which is responsive thereto to control operation of the decode unit.

At page 6, paragraph 9, the Office Action rejected claim 5, asserting, "Song in view of Grochowski has further taught that when the emulation unit is in a precise watch mode, it is operable to issue a request to the execution pipeline for guard resolution, the guard resolution being transmitted to the control circuitry of the emulation unit which is response thereto to control operation of the decode unit. Note from column 3, lines 25,35, of Grochowski, that a predicated instruction is prevented from executing until the guard is resolved...Therefore, when Song is in a precise watch mode, Grochowski has taught that it is beneficial to issue a request to the execution pipeline for guard resolution for the aforementioned reasons. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to issue a request to the execution pipeline for guard resolution during a precise watch mode in Song." Applicants respectfully disagree with these assertions, as one of skill in the art would not have been motivated to combine Song and Grochowski in the manner asserted in the Office Action.

Specifically, Applicants disagree that one of skill in the art would have modified Song to issue a request to the execution pipeline for guard resolution. The architecture disclosed by Song

does not support predicated execution. Therefore, in Song it is impossible to perform resolution of guard values. Accordingly, the asserted combination of Song and Grochowski would not yield a functional system. Thus, claim 16 is patentable over the combination of Song and Grochowski.

Claims 17 and 18, newly added in this application, depend from claim 16 and are patentable for at least the same reasons.

CONCLUSION

In view of the foregoing amendments and remarks, this application should now be in condition for allowance. A notice to this effect is respectfully requested. If the Examiner believes, after this amendment, that the application is not in condition for allowance, the Examiner is requested to call the Applicant's attorney at the telephone number listed below.

If this response is not considered timely filed and if a request for an extension of time is otherwise absent, Applicant hereby requests any necessary extension of time. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 23/2825.

Respectfully submitted,

Andrew Cofler et al.

By: _____

James H. Morris, Reg. No. 46,066
WOLF, GREENFIELD & SACKS, P.C.
600 Atlantic Avenue
Boston, Massachusetts 02210
Tel. (617) 720-3500

Attorney's Docket No.: S1022.80583US00
Dated: November 22, 2004